

Progetto di Basi di Dati

Sito per vendere e scaricare musica indie

1 - Raccolta delle specifiche della realtà d'interesse

Descrizione

Si vuole progettare una base di dati per la gestione di un piccolo **sito di articoli musicali digitali**.

L'accesso al sito avviene tramite la creazione di un "profilo utente", identificato univocamente da un username e per cui è necessario conoscere anche l'indirizzo e-mail in fase di registrazione. In seguito è possibile inserire facoltativamente un indirizzo (CAP, città, via e numero civico).

Un utente può registrarsi come un "privato" o come un "negozio".

Se ci si registra come "privato" occorre conoscere anche nome e cognome.

Se ci si registra come "negozio", invece, occorre conoscere numero di IBAN e numero di partita IVA del negozio e si ha un riepilogo semestrale dei guadagni.

Ogni utente può caricare sul sito uno o più "prodotti digitali", distinti univocamente dal "titolo". Di questi si vogliono memorizzare il prezzo, i formati digitali, i tag, la data di pubblicazione e la descrizione.

Gli utenti privati possono acquistare i prodotti caricati da altri utenti del sito, sia a pagamento che non, potendo poi recensire i prodotti acquistati con un voto e un breve commento. Di ogni recensione si vuole memorizzare la data d'invio della stessa, il voto, il testo scritto (se c'è) e il codice univoco identificativo. Ogni acquisto dispone di un codice univoco identificativo e di ogni acquisto si vogliono conoscere anche data e totale del pagamento.

Ogni prodotto può essere un "brano singolo" o "album" (non entrambi) o un "libro digitale". Dei primi occorre tener traccia della durata, dei secondi l'elenco delle tracce e la durata totale mentre dei terzi il numero di pagine.

Specifiche della realtà d'interesse

L'obiettivo della base di dati è quello di supportare una piccola piattaforma online per la diffusione e la vendita di musica indie. Piattaforme di questo tipo servono non solo per mettere a diretto contatto *creatori e consumatori*, ma anche per offrire a chi produce e chi vende una vetrina per i propri prodotti.

Il sito consente una registrazione gratuita scegliendo, al momento della creazione dell'account, se inserirsi nel sistema come utenti "privati" o come utenti "negozio". Gli utenti "negozio" hanno profili abilitati solo al caricamento e alla vendita di prodotti e solitamente sono utilizzati da gruppi di artisti che operano assieme. Gli utenti "privati" possono caricare e vendere prodotti come gli utenti "negozio" e, in più, hanno la

possibilità di acquistare e recensire i prodotti altrui. La distinzione nasce dall'idea che per un collettivo sia privo di interesse acquistare e recensire prodotti in gruppo perché sono sulla piattaforma esclusivamente per la vendita dei loro prodotti. Le persone facenti parte di un utente gruppo ("negozio") possono comunque registrarsi come utenti "privati". L'utente, che sia "negozio" o "privato", deve inserire in fase di registrazione un "username" e un indirizzo e-mail univoci. Gli utenti "negozio" necessitano inoltre di un codice IBAN e di una partita IVA per le loro vendite, mentre per gli utenti "privati" sono necessari dei dati anagrafici, quali nome e cognome. I negozi hanno un riepilogo dei guadagni semestrali.

I prodotti che possono essere caricati sono di due tipi: file musicali presentati come singoli brani o collezioni di tracce (queste chiamate album) e libri in formato digitale legati alla musica in vendita. Un prodotto in vendita può anche essere distribuito gratuitamente impostando il suo prezzo di vendita a 0,00 €. L'acquisizione di un prodotto gratuito per semplicità viene valutata comunque come un "acquisto". Tutti i prodotti devono essere in versione digitale con vari campi per essere descritti ai possibili acquirenti, sia sul piano del contenuto che del formato (titolo diverso da quello degli altri, media voti di tutte le recensioni, il prezzo in euro, i formati digitali, zero o più tag creati e assegnati da chi carica per facilitare la ricerca di prodotti simili e una descrizione testuale). I libri, inoltre, devono riportare il loro numero di pagine, i brani singoli la durata degli stessi e gli album l'elenco delle tracce assieme alla durata totale.

I "tag" menzionati poc'anzi consistono in brevi parole chiave che vengono associate ai prodotti da chi li ha caricati. Ogni prodotto può avere al massimo 5 "tag", diversi fra loro e concepiti per essere utili agli utenti che vogliono cercare un prodotto in base a delle categorie, come il genere musicale, gli strumenti usati, i temi, ecc. Altrimenti, gli utenti possono cercare i prodotti caricati nel database in base al titolo e ai nomi di chi li ha caricati con la possibilità di ordinare successivamente i risultati di queste ricerche in ordine alfabetico secondo il titolo o l'autore del prodotto. Inoltre, è possibile ordinare i risultati delle ricerche anche in base al numero di pagine, alla data di caricamento, alla media dei voti o al prezzo.

Gli utenti "privati" hanno, inoltre, la possibilità di acquistare uno o più prodotti in un unico acquisto. Degli acquisti si vuole memorizzare la data di quando sono stati finalizzati e associare a ciascuno di essi un codice univoco identificativo generato dal sistema.

Un utente "privato" che ha acquistato un prodotto può scrivere una "recensione" su di esso. Una "recensione" consiste in un voto espresso con interi da 1 a 5 e può includere un breve commento. Non è possibile per un utente scrivere più recensioni per lo stesso prodotto. Le "recensioni", inoltre, possono essere ordinate cronologicamente quando visualizzate dagli utenti.

Se un utente “negozio” decide di rimuovere il proprio profilo dal sistema allora non sarà più possibile per gli altri utenti trovare i prodotti messi in vendita dall’utente “negozio”, né sarà possibile trovare l’utente “negozio” stesso, ma resteranno salvati gli acquisti dei relativi prodotti finalizzati con gli utenti ancora presenti nella base di dati che li hanno effettuati. Se invece è un utente “privato” a rimuovere il suo profilo, allora le recensioni da lui scritte resteranno ma riporteranno come nome di default “utente cancellato”.

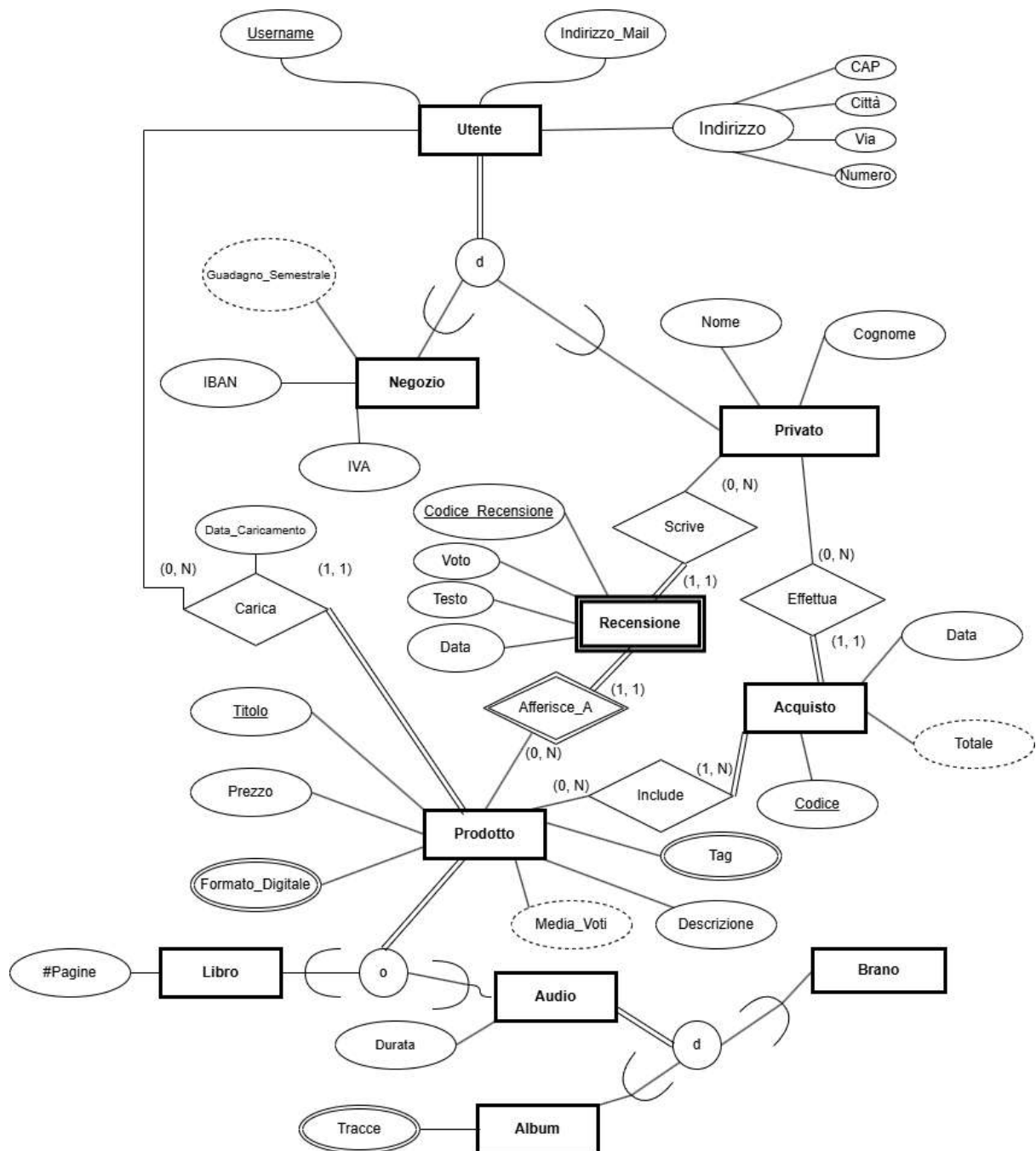
Glossario dei termini

Termine	Significato
Utente	Profilo creato e gestito da un singolo individuo o da un gruppo. Necessario per accedere al sito.
Privato	Utente che si è registrato come singolo individuo. Può caricare i suoi prodotti per renderli disponibili alla vendita agli altri utenti e può acquistare e recensire quelli altrui.
Negozi	Utente che si è registrato come gruppo. Può caricare e vendere i propri prodotti e deve avere una partita IVA e un codice IBAN in previsione di un rilevante numero di vendite. Può visualizzare la somma dei guadagni dei sei mesi precedenti.
Recensione	Voto numerico intero da 1 a 5 accompagnato eventualmente da un breve commento. Può essere scritta per un singolo prodotto da un utente che lo ha acquistato. Sono visibili a tutti e viene calcolata e mostrata la media per i singoli prodotti.
Acquisto	Operazione con la quale un utente “privato” ottiene uno o più prodotti (anche gratuitamente).
Prodotto	Opera digitale distribuita gratuitamente o a pagamento. Può consistere in un audio e un libro o uno solo di essi.
Libro	Prodotto e-book a tema musicale o affine agli altri prodotti.
Audio	Brano singolo o album
Album	Prodotto consistente in una collezione di tracce musicali in formato digitale.
Brano	Prodotto consistente in una singola traccia musicale.

2 - Progettazione concettuale della base di dati

Schema EER

Lo schema EER che si ottiene procedendo con la progettazione concettuale della base di dati è il seguente:



Dizionario delle entità

Legenda: sotto-entità, attributo multivalore, attributo ridondante, attributo composto, entità debole, chiave candidata

Entità	Descrizione	Attributi	Identificatore
Utente	Profilo creato e gestito da una o più persone	<ul style="list-style-type: none"> - Username - Indirizzo_Mail - Indirizzo 	Username
Privato	Utente che carica, vende, acquista e recensisce prodotti	<ul style="list-style-type: none"> - Nome - Cognome 	-
Negozi	Utente che carica e vende i propri prodotti	<ul style="list-style-type: none"> - IBAN - IVA - Guadagno_Semestrale 	-
Acquisto	Operazione con la quale un "privato" ottiene uno o più prodotti	<ul style="list-style-type: none"> - Codice - Data - Totale 	Codice
Recensione	Voto e commento di un "privato" ad un prodotto	<ul style="list-style-type: none"> - Data - Testo - Voto - Codice_Recensione 	Codice_Recensione
Prodotto	Libro e Audio o solo uno dei due.	<ul style="list-style-type: none"> - Titolo - Descrizione - Media_Voti - Tag - Prezzo - Formato_Digitale 	Titolo
Libro	Prodotto e-book	<ul style="list-style-type: none"> - #Pagine 	-
Audio	Prodotto Album o Brano	<ul style="list-style-type: none"> - Durata (in secondi) 	
Album	Prodotto Audio consistente in una collezione di tracce musicali	<ul style="list-style-type: none"> - Tracce 	-
Brano	Prodotto Audio consistente in un singolo brano		-

Dizionario delle relazioni

Relazione	Descrizione	Entità coinvolte	Attributi
Scrive	Un "privato" scrive una recensione relativa a un prodotto da lui acquistato	Privato (0, N) Recensione (1, 1)	-
Effettua	Un "privato" effettua l'acquisto di uno o più prodotti	Privato (0, N) Acquisto (1, 1)	-
Carica	Un utente può caricare più prodotti nel negozio	Utente (0, N) Prodotto (1, 1)	Data_Caricamento
Afferisce_A	Una recensione afferisce a un prodotto con un voto e un commento	Recensione (1, 1) Prodotto (0, N)	-
Include	Un acquisto include almeno un prodotto, anche se gratuito	Acquisto (1, N) Prodotto (0, N)	-

Vincoli non esprimibili nello schema

Sono inoltre presi in considerazione i seguenti **vincoli**, oltre il deducibile dallo schema EER:

- L'attributo "Guadagno_Semestrale" dell'entità "Negozio" si ottiene dalla somma di tutti i "Prezzi" dei "Prodotti" caricati dal "Negozio" e acquistati da altri nei sei mesi precedenti a quello corrente.
- L'attributo "Voto" dell'entità "Recensione" deve essere un numero intero compreso fra 1 e 5.
- L'attributo "Testo" dell'entità "Recensione" deve avere un numero di caratteri minore o uguale a 700.
- L'attributo "Totale" dell'entità "Acquisto" si ottiene dalla somma dei valori di tra "Prezzo" di tutte le entità "Prodotto" incluse nell' "Acquisto".
- L'attributo "Media_Voti" dell'entità "Prodotto" si ottiene dalla media aritmetica dei "Voti" delle "Recensioni" che afferiscono al "Prodotto" mostrando fino a una cifra decimale.

- L'attributo "Descrizione" dell'entità "Prodotto" deve avere un numero di caratteri minore o uguale di 5000.
- L'attributo "#Pagine" dell'entità "Libro" deve avere avere un intero positivo maggiore di 0.
- L'attributo "Tracce" dell'entità "Album" deve avere almeno 2 valori.
- L'attributo multivalore "Tag" dell'entità "Prodotto" può avere massimo 5 valori di massimo 15 caratteri l'uno.

3 - Definizione delle procedure per la gestione della base di dati

Tavola dei volumi

Di seguito, è definita la tavola dei volumi della base di dati:

Concetto	Tipo	Carico Applicativo
Utente	E	1000
Privato	E	750
Negozio	E	250
Acquisto	E	1500
Recensione	E	600
Prodotto	E	1500
Libro	E	300
Audio	E	1300
Album	E	900
Brano	E	300
<i>Scriva</i>	R	600
<i>Effettua</i>	R	1500
<i>Carica</i>	R	1500
<i>Afferisce_A</i>	R	600
<i>Include</i>	R	3000

Tavola delle operazioni

Di seguito la tavola delle operazioni per la gestione dei dati memorizzati nella base di dati:

N°	Operazione	Tipo	Frequenza
1	Creare un nuovo utente “privato”	I	15 / mm
2	Creare un nuovo utente “negozio”	I	5 / mm
3	Caricare un audio	I	15 / mm
4	Caricare un libro	I	9 / mm
5	Caricare un prodotto libro e album	I	6 / mm
6	Effettuare un acquisto	I	30 / mm
7	Scrivere una recensione	I	12 / mm
8	Visualizzare tutti i gli acquisti effettuati da un privato con i rispettivi totali, ordinati con data decrescente	I	10 / mm
9	Visualizzare tutti i prodotti caricati da un utente con prezzo inferiore a 20,00 € e una media voti maggiore di 4.0	I	50 / mm
10	Visualizzare tutti i prodotti con un dato tag, ordinati per prezzo in ordine crescente	I	60 / mm
11	Visualizzare tutti gli audio caricati da un utente	I	10 / mm
12	Visualizzare le recensioni di un prodotto in ordine temporale decrescente	I	50 / mm
13	Visualizzare il guadagno semestrale di un negozio	B	2 / aa
14	Selezionare il prodotto più acquistato negli ultimi 6 mesi	B	2 / aa

4 - Progettazione Logica

Analisi delle ridondanze

Nello schema ci sono i seguenti dati ridondanti: l'attributo Guadagno_Semestrale dell'entità Negozio, l'attributo Totale dell'entità Acquisto e l'attributo Media_Voti dell'entità Recensione.

Guadagno_Semestrale lo si può ottenere sommando i valori dell'attributo Prezzo dei vari Prodotti caricati da un singolo Negozio limitandosi a quelli acquistati negli ultimi sei mesi. Memorizzato come mediumint, occupa 3 byte moltiplicati per i 250 Negozi per un totale di 750 byte.

Totale deriva dalla somma dei valori Prezzo dei singoli Prodotti inclusi in un Acquisto. Memorizzato come smallint, occupa 2 byte moltiplicati per i 1500 Acquisti per un totale di 3000 byte (3 KB).

Media_Voti è dato dalla media aritmetica dell'attributo Voto delle Recensioni che afferiscono ad un singolo Prodotto. Memorizzato come tinyint, occupa 1 byte moltiplicato per 600 Recensioni per un totale di 600 byte.

Analisi ridondanza Totale

Op 6 con ridondanza				Op 6 senza ridondanza			
Nome	Tipo	Occorrenze	Accesso	Nome	Tipo	Occorrenze	Accesso
Acquisto	E	1	S	Acquisto	E	1	S
Include	R	2	S	Include	R	2	S
Prodotto	E	2	L	Prodotto	E	2	L
TOT: (3S+2L)*30/mm = 240 accessi mensili				TOT: (3S+2L)*30/mm = 240 accessi mensili			

Sull'operazione 6 la ridondanza non impatta perché il valore Totale è salvato in Acquisto e questa entità verrà sempre aperta in scrittura.

Op 8 con ridondanza				Op 8 senza ridondanza			
Nome	Tipo	Occorrenze	Accessi	Nome	Tipo	Occorrenze	Accessi
Privato	E	1	L	Privato	E	1	L
Effettua	R	0,04	L	Effettua	R	0,04	L
Acquisto	E	0,04	L	Acquisto	E	0,04	L
				Include	R	0,08	L
				Prodotto	E	0,08	L
TOT: (1,08L)*10/mm = 10,8 accessi mensili				TOT: (1,24L)*10/mm = 12,4 accessi mensili			

In conclusione, eliminando la ridondanza si risparmierebbero 3 KB al costo di 2,4 accessi mensili in più. Pur essendo entrambi valori piccoli, 2,4 accessi mensili in più peserebbero di meno. Si proseguirà rimuovendo la ridondanza perché l'operazione viene usata con bassa frequenza.

Analisi ridondanze Guadagno_Semestrale

Per l'Op 6 si considera che in media ogni acquisto includa 2 prodotti caricati da negozi distinti.

Op 6 con ridondanza				Op 6 senza ridondanza			
Nome	Tipo	Occorrenze	Accessi	Nome	Tipo	Occorrenze	Accessi
Acquisto	E	1	S	Acquisto	E	1	S
Include	R	2	S	Include	R	2	S
Prodotto	E	2	L	Prodotto	E	2	L
Carica	R	2	L				
Negozi	E	2	S				
TOT: $(5S+4L)*30/\text{mm} = 420$ accessi mensili				TOT: $(3S+2L)*30/\text{mm} = 240$ accessi mensili			

Per l'Op13 si considera che ogni negozio vende in media 8 prodotti l'anno.

Op 13 con ridondanza				Op 13 senza ridondanza			
Nome	Tipo	Occorrenze	Accessi	Nome	Tipo	Occorrenze	Accessi
Negozi	E	1	L	Negozi	E	1	L
				Carica	R	4	L
				Prodotto	E	4	L
				Include	R	8	L
TOT: $(1L)*0,17/\text{mm} = 0,028$ accessi mensili				TOT: $(17L)*0,17/\text{mm} = 2,89$ mensili			

Nonostante l'Op13 senza ridondanza comporti effettivamente più accessi, è un'operazione batch e conviene eliminare la ridondanza considerata la differenza di accessi dell'Op6 con e senza ridondanza, anche perché l'attributo pesa 750 byte.

Analisi ridondanze Media_Voti

Op 7 con ridondanza				Op 7 senza ridondanza			
Nome	Tipo	Occorrenze	Accessi	Nome	Tipo	Occorrenze	Accessi
Recensione	E	1	S	Recensione	E	1	S
Afferisce_A	R	1	S	Afferisce_A	R	1	S
Prodotto	E	1	S	Prodotto	E	1	L
TOT: (3S)*12/mm = 72 accessi mensili				TOT: (2S+1L)*12/mm = 60 accessi mensili			

Per l'Op9 si considera che non abbia senso visualizzare tutti caricati da un utente che ne ha caricati meno di 2 senza recensioni.

Op 9 con ridondanza				Op 9 senza ridondanza			
Nome	Tipo	Occorrenze	Accessi	Nome	Tipo	Occorrenze	Accessi
Utente	E	1	L	Utente	E	1	L
Carica	R	2	L	Carica	R	2	L
Prodotto	E	2	L	Prodotto	E	2	L
				Afferisce_A	R	2	L
				Recensione	E	2	L
TOT: (5L)*50/mm = 250 accessi mensili				TOT: (9L)*50/mm = 450 accessi mensili			

Mantenendo la ridondanza si avrebbero in totale 322 accessi mensili con 600 byte. Eliminandola invece 510 accessi.

In questo caso conviene mantenere la ridondanza anche perché 600 byte sono meno significativi dei 750 e 3000 degli altri due attributi ridondanti.

Eliminazione delle gerarchie ed eliminazione degli attributi multi-valore e composti

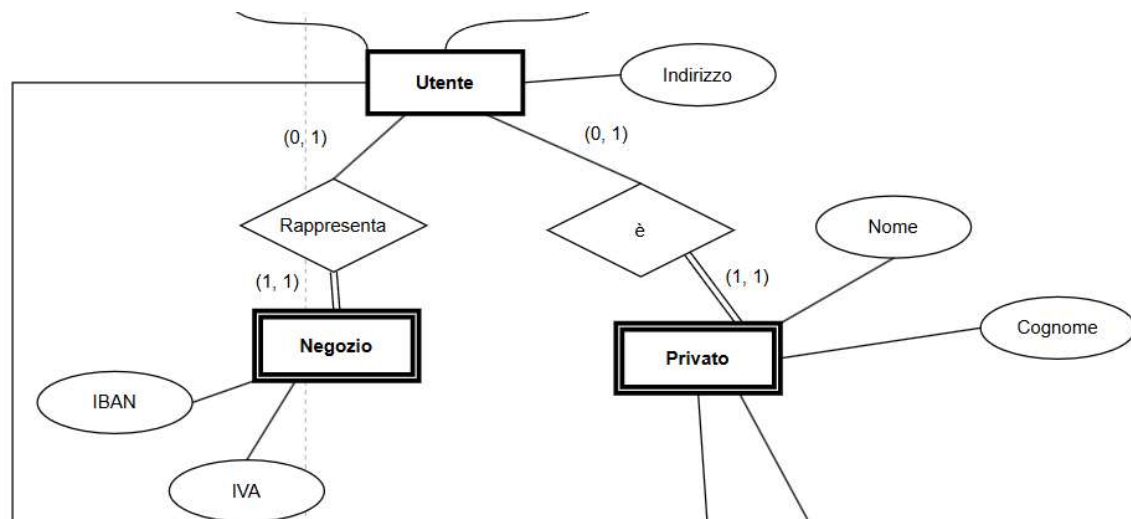
(Le modifiche sono presentate assieme in base alle aree del diagramma dall'alto in basso)

Per quanto riguarda **le sotto-classi di Utente**, ovvero **Negozi** e **Privato**, si è deciso di renderle entità deboli di **Utente**. In questo modo si evitano gli attributi nulli che ci sarebbero stati accorpendo le sotto-classi alla super-classe.

Questo richiede l'aggiunta del seguente vincolo:

- un **Utente** non può partecipare sia alla relazione **Rappresenta** che a quella **è** ma deve partecipare ad una delle due

L'**attributo composto Indirizzo** diventa un unico attributo per contenere una stringa di caratteri.



Le **sotto-classi Brano e Album** della super-classe **Audio** sono state accorpate non partecipando a relazioni specifiche e non avendo molti attributi.

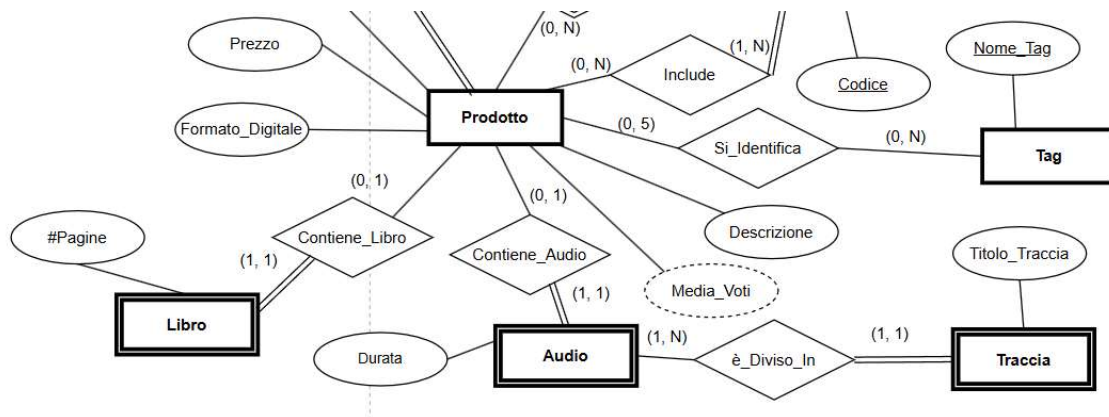
Per preservare l'**attributo multi-valore Tracce** si è scelto di trasformarlo in entità debole di **Audio** con la possibilità di inserire come unico **Titolo_Traccia** il nome di un eventuale singolo **Brano**.

Le **sotto-classi Libro e Audio** sono state convertite in entità deboli con le relazioni **Contiene_Libro** e **Contiene_Audio**. Si aggiunge il vincolo:

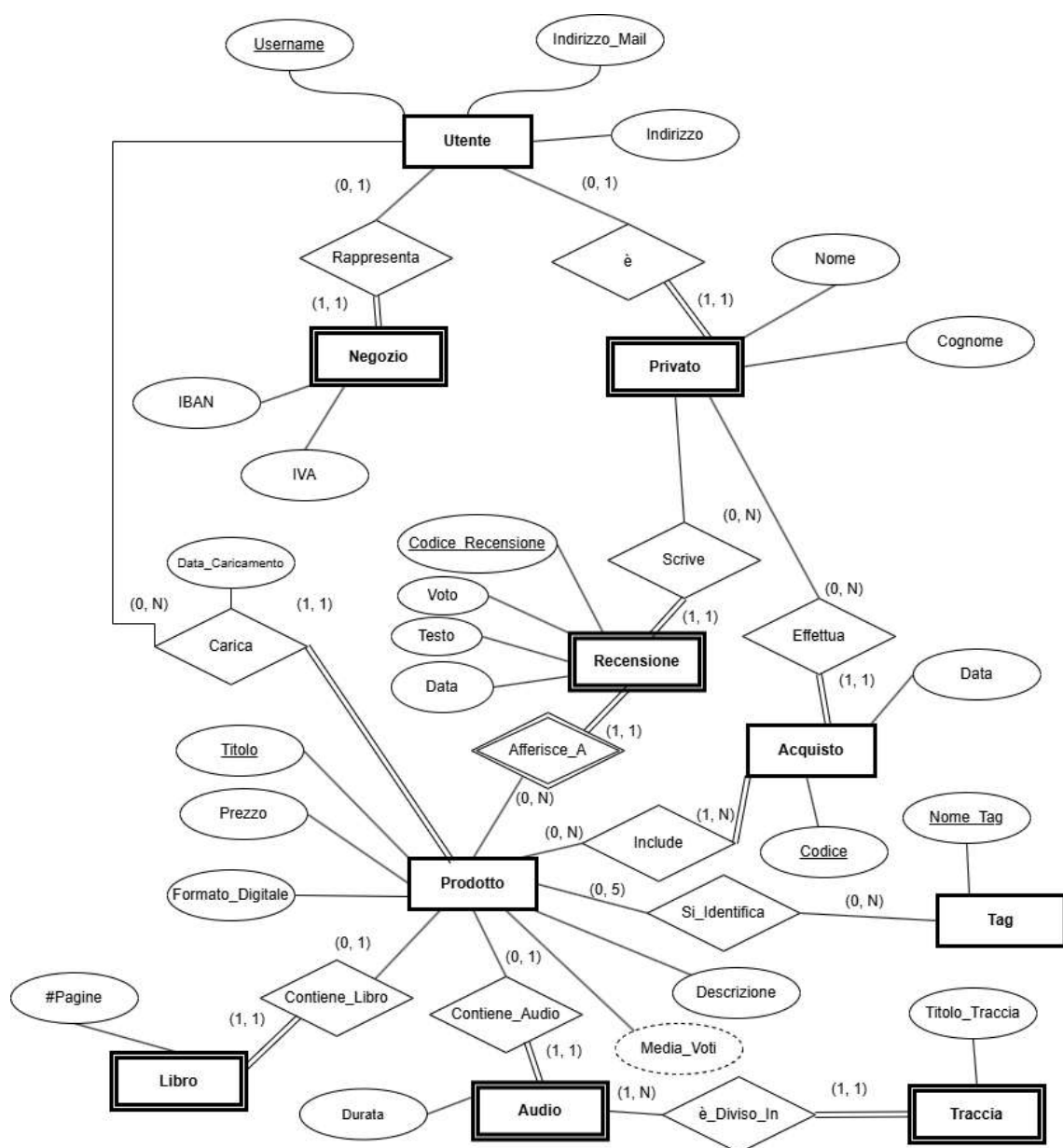
- un **Prodotto** deve partecipare almeno ad una delle relazioni **Contiene_Libro** e **Contiene_Audio**

L'**attributo multi-valore Formato_Digitale** è stato trasformato in un attributo stringa di caratteri.

L'**attributo multi-valore Tag** è stato convertito nell'entità **Tag** che partecipa alla relazione **Si_Identifica** di **Prodotto**. La sua chiave è il nome stesso del tag.



SCHEMA RISTRUTTURATO:



5 - Mapping

Utente(Username, Indirizzo_Mail, Indirizzo)
Negozio(IBAN, IVA, utente.Username↑)
Privato(Nome, Cognome, utente.Username↑)
Recensione(Codice_Recensione, utente.Username↑, prodotto.Titolo, Voto, Testo, Data)
Acquisto(Data, Codice, utente.Username↑)
Acquisti_Prodotti(acquisto.Codice↑, prodotto.Titolo↑)
Prodotto(Data_Caricamento, utente.Username↑, Titolo, Prezzo, Formato_Digitale, Descrizione, Media_Voti)
Tag(Nome_Tag)
Prodotto_Tag(prodotto.Titolo↑, tag.Nome_Tag↑)
Libro(prodotto.Titolo↑, pagine)
Audio(prodotto.Titolo↑, durata)
Traccia(prodotto.Titolo↑, Titolo_Traccia)

Normalizzazione

Lo schema relazionale mappato non richiede modifiche per essere normalizzato fino alla 3NF.

1NF: rispettata perché gli attributi sono atomici come conseguenza dell'eliminazione degli attributi composti e multi-valore.

2NF: rispettata perché non sono presenti dipendenze funzionali parziali rispetto a chiavi primarie.

3NF: rispettata perché sono assenti dipendenze transitive alle chiavi da attributi non chiave

6 - Database con MySQL

```
CREATE SCHEMA IF NOT EXISTS NegozioMusica;
USE NegozioMusica;

CREATE TABLE Utente (
    username VARCHAR(50) NOT NULL PRIMARY KEY,
    email VARCHAR(50) NOT NULL,
    indirizzo VARCHAR(100) NOT NULL
);

CREATE TABLE Negozio (
```

```

        username VARCHAR(50) NOT NULL PRIMARY KEY,
        IBAN VARCHAR(34) NOT NULL,
        IVA VARCHAR(11) NOT NULL,
        CONSTRAINT fk_negozio_utente FOREIGN KEY(username)
            REFERENCES Utente(username) ON DELETE CASCADE
    );

CREATE TABLE Privato (
    username VARCHAR(50) NOT NULL PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    cognome VARCHAR(50) NOT NULL,
    CONSTRAINT fk_privato_utente FOREIGN KEY(username)
        REFERENCES Utente(username) ON DELETE CASCADE
);

CREATE TABLE Prodotto (
    data_caricamento TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP(),
    username VARCHAR(50) NOT NULL,
    titolo VARCHAR(50) NOT NULL PRIMARY KEY,
    prezzo FLOAT NOT NULL,
    formato_digitale VARCHAR(100) NOT NULL,
    descrizione VARCHAR(5000) NOT NULL CHECK(LENGTH(descrizione)
<= 5000),
    media_voti FLOAT(2,1) NOT NULL DEFAULT 0,
    CONSTRAINT fk_prodotto_utente FOREIGN KEY(username)
        REFERENCES Utente(username) ON DELETE CASCADE
);

CREATE TABLE Recensione(
    codice_recensione SERIAL PRIMARY KEY, -- Alias di BIGINT
UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE
    username VARCHAR(50) NOT NULL,
    titolo VARCHAR(50) NOT NULL,
    voto INT NOT NULL CHECK(voto >= 0 AND voto <= 5),
    testo VARCHAR(700) NOT NULL CHECK(LENGTH(testo) <= 700)
DEFAULT '',
    data_recensione TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP(),
    CONSTRAINT fk_recensione_privato FOREIGN KEY(username)
        REFERENCES Privato(username) ON DELETE CASCADE
);

```

```
CREATE TABLE Acquisto(  
    codice_acquisto SERIAL PRIMARY KEY, -- Alias di BIGINT  
    UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE  
    username VARCHAR(50) NOT NULL,  
    data TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP(),  
    CONSTRAINT fk_acquisto_utente FOREIGN KEY(username)  
        REFERENCES Utente(username) ON DELETE CASCADE  
);  
  
CREATE TABLE Acquisti_Prodotti(  
    codice_acquisto BIGINT UNSIGNED NOT NULL,  
    titolo VARCHAR(50) NOT NULL,  
    PRIMARY KEY(codice_acquisto, titolo),  
    CONSTRAINT fk_acquisti_prodotti_acquisto FOREIGN  
KEY(codice_acquisto)  
        REFERENCES Acquisto(codice_acquisto) ON DELETE CASCADE,  
    CONSTRAINT fk_acquisti_prodotti_prodotto FOREIGN KEY(titolo)  
        REFERENCES Prodotto(titolo) ON DELETE CASCADE  
);  
  
CREATE TABLE Tag(  
    nome_tag VARCHAR(15) NOT NULL PRIMARY KEY  
    CHECK(LENGTH(nome_tag) <= 15)  
);  
  
CREATE TABLE Prodotto_Tag(  
    titolo VARCHAR(50) NOT NULL,  
    nome VARCHAR(50) NOT NULL,  
    PRIMARY KEY(titolo, nome),  
    CONSTRAINT fk_prodotto_tag_prodotto FOREIGN KEY(titolo)  
        REFERENCES Prodotto(titolo) ON DELETE CASCADE,  
    CONSTRAINT fk_prodotto_tag_tag FOREIGN KEY(nome)  
        REFERENCES Tag(nome_tag) ON DELETE CASCADE  
);  
  
CREATE TABLE Libro(  
    titolo VARCHAR(50) NOT NULL PRIMARY KEY,  
    pagine INT NOT NULL CHECK(pagine > 0),  
    CONSTRAINT fk_libro_prodotto FOREIGN KEY(titolo)  
        REFERENCES Prodotto(titolo) ON DELETE CASCADE  
);
```



```
CREATE TABLE Audio(  
    titolo VARCHAR(50) NOT NULL PRIMARY KEY,  
    durata INT NOT NULL,  
    CONSTRAINT fk_audio_prodotto FOREIGN KEY(titolo)  
        REFERENCES Prodotto(titolo) ON DELETE CASCADE  
);
```

```
CREATE TABLE Traccia(  
    titolo VARCHAR(50) NOT NULL PRIMARY KEY,  
    titolo_traccia VARCHAR(50) NOT NULL,  
    CONSTRAINT fk_traccia_audio FOREIGN KEY(titolo)  
        REFERENCES Audio(titolo) ON DELETE CASCADE  
);
```

```
DELIMITER $$  
CREATE TRIGGER tr_media_voti  
    AFTER INSERT ON Recensione  
    FOR EACH ROW  
BEGIN  
    DECLARE media FLOAT(2,1);  
    SELECT AVG(voto) INTO media FROM Recensione WHERE titolo =  
NEW.titolo;  
    UPDATE Prodotto SET media_voti = media WHERE titolo =  
NEW.titolo;  
END $$  
DELIMITER ;
```

```
DELIMITER $$  
CREATE TRIGGER tr_max_tags  
    BEFORE INSERT ON Prodotto_Tag  
    FOR EACH ROW  
BEGIN  
    DECLARE count_tags INT;  
    SELECT COUNT(*) INTO count_tags FROM Prodotto_Tag WHERE titolo  
= NEW.titolo;  
    IF count_tags > 5 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Troppi tag per questo prodotto';  
    END IF;  
END $$
```

```
DELIMITER ;
```

7 - Implementazione query SQL

Operazione 1:

```
INSERT INTO Utente (username, email, indirizzo)
VALUES('Steffy', 'stefo@mail.it', 'via Roma 9, Avellino');
INSERT INTO Privato (username, nome, cognome)
VALUES('Steffy', 'Stefano', 'Ligure');
```

Operazione 2:

```
INSERT INTO Utente (username, email, indirizzo)
VALUES('Murko', 'murko@mail.it', '00199 Roma Via Concordia
11');
INSERT INTO Negozio (username, IBAN, IVA)
VALUES('IT09F0300203280625926433964', '86364109827');
```

Operazione 3:

```
INSERT INTO Prodotto(username, titolo, prezzo, formato_digitale,
descrizione)
VALUES('Murko', 'Dark Dance', 5.00, 'mp3', 'Musica ambient
basata sul folclore Svedese');
INSERT INTO Audio(titolo, durata)
VALUES('Dark Dance', 500);
INSERT INTO Traccia(titolo, titolo_traccia)
VALUES('Dark Dance', 'Dark dance traccia');
INSERT INTO Tag(nome_tag)
VALUES('dark');
INSERT INTO Tag(nome_tag)
VALUES('nord');
```

Operazione 4:

```
INSERT INTO Prodotto (username, titolo, prezzo, formato_digitale,
descrizione)
VALUES('Murko', 'Nord Oscuro', 2.00, 'epub', 'Companion del
brano Dark Dance');
INSERT INTO Libro (titolo, pagine)
VALUES('Nord Oscuro', 25);
```

```
INSERT INTO Tag (nome_tag)
VALUES('dark');
```

Operazione 5:

```
INSERT INTO Prodotto (username, titolo, prezzo, formato_digitale,
descrizione)
VALUES ('Steffy', 'Rombi', 0.00, 'epub;mp3', '?')
INSERT INTO Libro (titolo, pagine)
VALUES ('Rombi', 23);
INSERT INTO Audio (titolo, durata)
VALUES (200);
```

Operazione 6:

```
INSERT INTO Prodotto(username, titolo, prezzo, formato_digitale,
descrizione)
VALUES('utente1', 'Il Signore degli Anelli', 10.0, 'pdf', 'Il
Signore degli Anelli è un romanzo fantasy scritto da J.R.R.
Tolkien');
INSERT INTO Acquisto(username) VALUES('Ciccio');
INSERT INTO Acquisti_Prodotti(codice_acquisto, titolo) VALUES(1,
'Il Signore degli Anelli');
```

Operazione 7:

```
INSERT INTO Recensione(username, titolo, voto, testo)
VALUES('tendis', 'Dark Dance', 5, 'Il miglior libro che abbia
mai letto');
```

Operazione 8:

```
SELECT A.codice_acquisto, SUM(P.prezzo) AS Somma FROM
Acquisti_Prodotti
JOIN Acquisto A on Acquisti_Prodotti.codice_acquisto =
A.codice_acquisto
JOIN Prodotto P on Acquisti_Prodotti.titolo = P.titolo
WHERE A.username = 'Marta'
GROUP BY A.codice_acquisto, A.data
ORDER BY A.data DESC;
```

Operazione 9:

```
SELECT titolo FROM Prodotto WHERE username = 'Ciccio'
```

```
AND prezzo < 20;
```

Operazione 10:

```
SELECT P.titolo FROM Tag
  JOIN Prodotto_Tag PT ON Tag.nome_tag = PT.nome
  JOIN Prodotto P ON PT.titolo = P.titolo
 WHERE Tag.nome_tag = 'Rock'
 ORDER BY P.prezzo ASC;
```

Operazione 11:

```
SELECT A.titolo FROM Audio
  JOIN Predo.Audio A on Audio.durata = A.durata
  JOIN Prodotto P on A.titolo = P.titolo
 WHERE username = 'Marta';
```

Operazione 12:

```
SELECT R.codice_recensione, R.voto, R.testo FROM Recensione R
  JOIN Prodotto P on R.titolo = P.titolo
 WHERE P.titolo = 'Dieciii'
 ORDER BY R.data_recensione DESC;
```

Operazione 13:

```
SELECT SUM(prezzo) FROM Acquisti_Prodotti
  JOIN Prodotto P on Acquisti_Prodotti.titolo = P.titolo
  JOIN Acquisto A on Acquisti_Prodotti.codice_acquisto =
A.codice_acquisto
 WHERE P.username = 'Ciccio' AND A.data > CURRENT_DATE() -
INTERVAL(6) MONTH;
```

Operazione 14:

```
SELECT P.titolo, COUNT(*) FROM Acquisti_Prodotti
  JOIN Prodotto P on Acquisti_Prodotti.titolo = P.titolo
  JOIN Acquisto A on Acquisti_Prodotti.codice_acquisto =
A.codice_acquisto
 WHERE A.data > CURRENT_DATE() - INTERVAL(6) MONTH
 GROUP BY P.titolo
 ORDER BY COUNT(*) DESC
 LIMIT 1;
```

ESEMPI DI ESECUZIONE QUERY

Op 8: vengono mostrati tutti gli acquisti effettuati dall'utente privato Marta con la colonna codice di acquisto ed il prezzo totale dei singoli prodotti in ognuno di essi.

```
1 • SELECT A.codice_acquisto, SUM(P.prezzo) AS Somma FROM Acquisti_Prodotti
2     JOIN Acquisto A on Acquisti_Prodotti.codice_acquisto = A.codice_acquisto
3     JOIN Prodotto P on Acquisti_Prodotti.titolo = P.titolo
4     WHERE A.username = 'Marta'
5     GROUP BY A.codice_acquisto, A.data
6     ORDER BY A.data DESC;
```

code_acquisto	Somma
1	49.97999954223633
6	69.98999786376953

Op 10 Vengono mostrati tutti i prodotti associati al tag Rock mostrando una colonna con i loro titoli in ordine ascendente di prezzo.

```
63 • SELECT P.titolo FROM Tag
64     JOIN Prodotto_Tag PT ON Tag.nome_tag = PT.nome
65     JOIN Prodotto P ON PT.titolo = P.titolo
66     WHERE Tag.nome_tag = 'Rock'
67     ORDER BY P.prezzo ASC;
```

titolo
Balls
Musica Blu
Time
Hole
NineOne